

GENCODE[™] TECHNIQUES FOR AUTHORS

by
William W. Tunnicliffe
Vice President, Information Technologies
Graphics Communication Association
Arlington, Virginia

This presentation is addressed to the needs and desires of the author. Hopefully, we have oriented the structure and sequence with that consideration in mind. Based on some of the things that I've heard earlier this week, the author may get his creative impulse in his study, in his recliner, in a Philadelphia-type study, at his desk, in his publisher's foyer, but, basically, what is he after? He's after content, and you're going to hear me emphasize this quite a bit in the course of the day. Essentially, it is the *quest for quality*. If you've been reading airplane magazines, you'll see that *Philadelphia Magazine* had a quest for quality, in which they triumphed: fifty awards for editorial excellence. The author might think of using a typewriter, but this week, as you note from *The Philadelphia Inquirer*, typewriters have been given up. They won't be manufacturing any more manual models. You've heard allusions to, or other comments about, word processors and microcomputers, so it is more likely that he will pass the typewriter by, think of word processors, such as a WANG or an EXXON, or he may take a designer's terminal. More probably, based on the numbers you are about to see, it may be a garret outfitted with a wood stove and some kind of a microcomputer terminal with, hopefully, some degree of intelligence—a suitable environment and an opportunity to put down what he has in mind.

Chart 1 displays the statistics projected by Byte Magazine for microcomputer shipments on a worldwide basis. You will see that 3 million of these should be shipped in 1984 and that the trend is prospectively in that continuing direction.

Chart 2 displays the personal shipments in the United States itself. You will see that the numbers are accumulating on both the home and the business front. We make an *a priori* assumption that these are the two cases of interest to us. Granted that there are authors in education, but the premise above is made on the basis that those in educational institutions have access through their terminals to more powerful computer systems than the microcomputers themselves.

Chart 3 provides an interesting comparison because it relates to installed microcomputers as against the shipments in any particular area or particular category. The particularly interesting thing is that you see that, about halfway through 1984, over 5 million microcomputers will be in business and perhaps some 2½ million in the home. This is a number which I would assume will exert some pressure, author pressure, if you like, as opposed to peer pressure or publisher pressure or any of the other pressures that you've heard discussed earlier.

What does he have for the microcomputer? They are certainly available (Figure 2), and the prices are coming down into a range of accommodation. They're convenient, and they can be most effective. As to which one of these is going to be dominant, the current ads in just one flight magazine suggest that "dominance" will continue to be a relative matter: Apple, Compaq, DEC, and IBM—IBM/Japan for those of you who attended Lufkin Productions' seance earlier this week on the Lord High Systems Analyst. This seance, if you like, could be called the Lord High Systems Synthesizer, since that's the exercise which you'll be going through as you embark upon this selection. Hyperion, Phillips, Radio Shack, TI, and, of course, we couldn't leave Ben (Franklin) out.

There are many roads to Main Frame City, and IBM and Apple—joined via Quadram—are symbolic of the fact that you can get a microcomputer, you can tie on so that you can create your manuscript off-line on your own system, and you then can give it to whomsoever you may have in mind. You can go to your friendly typesetter, and many of these have interfacing capabilities, or, if you reflect on this for a moment, you may think two things. First, that you don't want to get tied to particular codes, whether they're typesetting codes or element identifier codes—a matter we shall develop further. You still may not want to get tied to one particular system—you may have manuscript for which the author has not yet selected the publisher, or the situation in which the publisher has not yet selected the author.

The author should certainly be concerned with **generic coding** (Figure 3) and what we call **transportable files**, as should a publisher for reasons that we will develop. For those of us who are long in the tooth in this field—and I am pleased to see in the audience some of the people who have similarly carried this torch over the years, Sarah Rhodes in particular—I think it's fair to say (Figure 4) that we're convinced that generic coding will minimize cost, vulnerabilities, and aggravation. That last one is a not insignificant factor if you believe that the creative process should be a reasonably uninterrupted process.

As far as the author, the publisher, and the typesetter are concerned, the relationships over the years have largely been that there is one typesetter involved with the publisher for any given book. The symbology of the numbers in Figure 5 is that, if you use generic coding, you can shop around to various typesetters. If you're an author, those numbers can be transferred to a position between the author-publisher lines quite as well as between the publisher-typesetter lines. You can give a manuscript for review to different publishers and know that your coding can be utilized by

any one of the publishers who may accept your manuscript or with whom you would like to develop a particular relationship.

Who will lead you through the swamp? The Mountie or the Indian guide? That's one of the questions. And (Figure 6) what is generic coding? Each has his own definition, and this, if you like, is one brand. Generic coding requirements are really system requirements (Figure 7). One should note particularly that the requirements of methodology are highly disciplined. The requirements of detail are so flexible (they do not require, at the outset, the use of a cookbook) that their acceptance has been, shall we say, slower than would be desirable and they have taken different forms. Some forms involve typesetting codes, and I shall comment on the lack of desirability of mixing typesetting codes with editorially based generic codes.

Figure 8 depicts the process and, again, I will have hit the drum of content and form separation. You keep your source file on a content basis with data element identifiers. For conventional print-image publishing, you apply your typographic design through a table of typographic parameters, one set for each data element identifier. If you don't like what you see as the output image, you change the entries in the typographic parameter table ONLY, and then you rerun. You do NOT change the coding in the manuscript. Because this is a hook and eye situation, you can reach in with your cargo hooks and get any data element you wish. You can create your data base for the magnetic publishing endeavors *and/or* for any other use—in-process or output product—that you may have in mind: be that on the face of the terminal so that you can read it, on the line printer so that you can check it on hard copy, or over the wire on an inquiry basis so that you can sell it!

From the outset this has been a separation with the understanding that there is a preprocessor, an interpreter, an analysis phase, whatever language you choose to use, and that out of that you come to the composition processor for *whatever* visual output device there may be. It is not always—as you have seen—a photocomposition output.

Another evidence of the fact that you could use the content to come out to whatever form you've selected, you can then, if you have kept it separate, take the content and manipulate it for other purpose. You may excise portions for a shorter document. In a data publishing proposition you may select different data elements and put out different editions from the common data base—the classic example is the price list. You give one set of prices to the distributor, another to the wholesaler, another to the retailer, and yet another to the manufacturer himself for internal use, with ALL of the columns. Each list contains the common elements: description, part number, etc. and the list-particular data—i.e., the prices appropriate to that particular kind of list.

Meaning and representation! It is not, in our view, significant what bit configuration you assign to a particular character or to a particular control code. What is important is that it should be *known*. The author should tell the processor, (side sheet or header table) and it should be consistent. The underlying weave in this is **discipline in the editorial sense**, and, therefore, it must be consistent,

at least for this one manuscript, one document. You want to separate meaning and representation so that you can walk from machine to machine via translation tables (the “conversions” mentioned earlier). This is the input side.

On the output side, the reason for separation of form and content is to allow output-product diversity. You want to be able to design your output for different purposes at different times, and you want the flexibility to do this. It could be the large-type weekly of the *New York Times*. It could be large-type editions of *Readers' Digest*. It could be putting it out in Braille (and this is, in fact, done.) In order to emphasize the differences between the functions that are involved: if an author (Figure 9) becomes concerned with typography, during that moment or that period of time, he or she is *NOT the author, but rather the DESIGNER* thinking! In the capacity of *designer*, one pays attention to factors of style and form. There are those who have the abilities to consider this, and many publishers have their own typographic designers or their own staff of typographic designers and their own styles. The author should not waste any time becoming involved in that. No decisions about whether to put a thin space after the number for the footnote and such other things as they have been concerned with since the days of Ottmar Mergenthaler and should now expunge.

If we look further into this, we will see that there is a consideration at the moment of mixing output product and input content, but (Figure 10) we do not believe that this should be done at the stage of the manuscript. The concept of coexistent structures (Figure 11) is receiving a certain amount of discussion these days in the operational and standards environment. WYSIWYG and WYCIWYG relate to the fact that, as in word processors (Figure 12), *what you see is what you get*. If you delve behind the screen in the word processor, you often find that there are differences in machines. Some machines are control machines. Some machines are space machines. As you start to process this stuff, therein begins to lie the difficulty, the noise factor. Figure 13 illustrates the alternative: *what you code is what you get*. Ultimately, I believe, you will put your input in in coded form, that there will be firmware inside the word processor, and that the formatting for the word processor will be done by that firmware to create what you see on the screen. The coding, of course, then can be passed on directly in the transportable file for other purposes, e.g., typesetting.

The fundamentals of generic coding, therefore, include an analysis of your document, the planning for the entire process for that document, and keeping your structure classifications open-ended. You do not want a rigid situation unless you are working against a *document type* which has already been registered in some catalogue.

In the image form that you can get from this, you want to maintain typesetting independence, to retain the ability to change your parameters. You want composition independence to be able to change your page style and conditional-logic independence so that you can change the typographic design in terms of the space and arrangement of the different classes of the text. Tight linkage refers to the fact that every editorial element that you use must be identified, and there must be a specification for each and every editorial element that you've used.

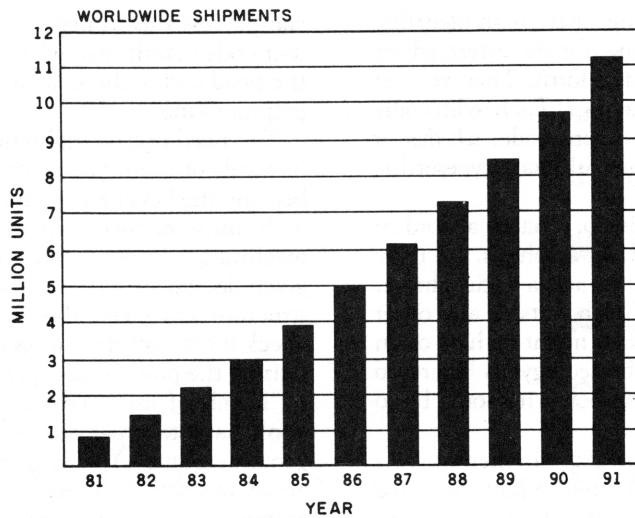


Chart 1 †

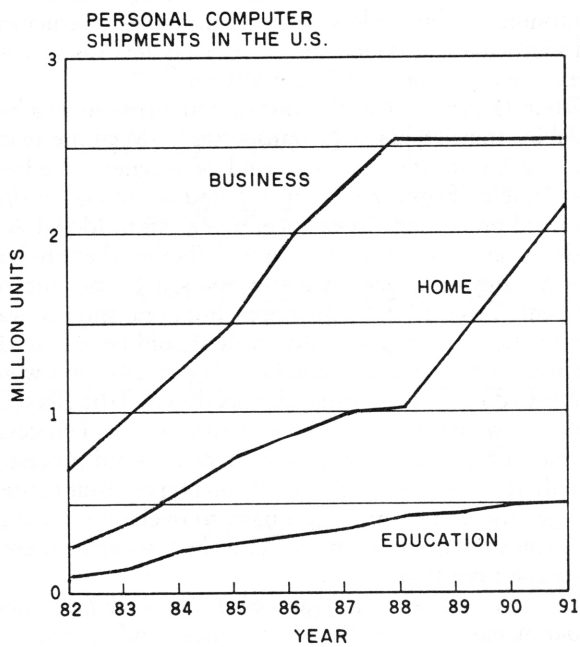


Chart 2

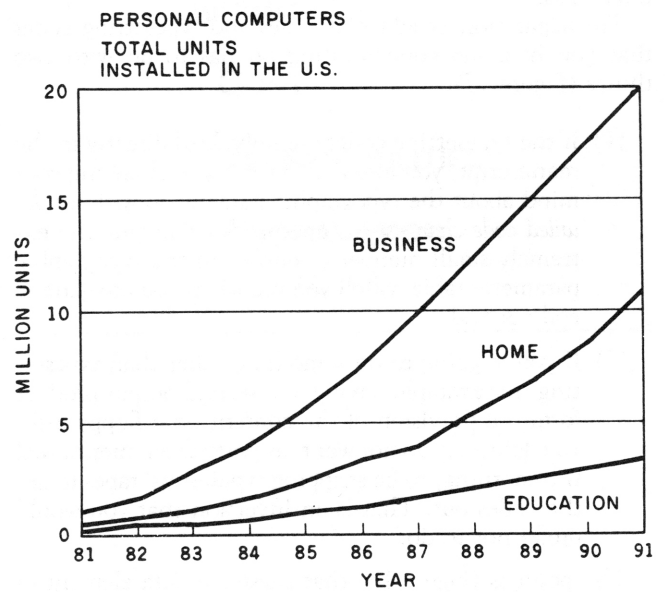


Chart 3

† From "Personal Computers in the Eighties" by Gregory S. Blundell appearing in the January, 1983 issue of *BYTE* magazine.
 Copyright © 1983 Byte Publications, Inc.
 Used with the permission of Byte Publications, Inc.

In a conventional chapter opening, there is an identifier for chapter line, for chapter title line, for the intermediate data about chairmen, speakers, and so forth. Then you get down to running text and to headings. Each editorially different element should have a different code, *whether or not* any two (or more) have the *same* ultimate typesetting when output.

Is Figure 14 italic? The answer is no. That is a word or words used for the purposes of editorial emphasis. Is Figure 15 italic? No. Rather, two book titles and a Latin phrase. If you want to make fog tests or Flesch tests or any other readability test, one of the dimensions might be how often does this author lapse into Latin phraseology? Is Figure 16 italic? No. This is a legal case citation and it should be so identified.

One of the things you must consider, whether you're a publisher or an author, is that you have to have analyzed (or synthesized) the editorial structure of the document under consideration. The structure may, and should be, open-ended. The initial structure can be defined by category—e.g., the assignment of editorial significance to a conventional I., A., 1., a., (1), (a), etc. structure and the assignment of expected constituent elements, e.g., paragraphs, lists, etc. Generic coding is based upon rules of logic and follows the express (or implicit) logic of the editorial structure and related content. This requires, of course, that the editorial structure must itself be logical—which it ordinarily is.

The separation of editorial codes and typesetting codes that you hear me concentrating on has to do with two things (Figure 17):

- (1) If the typesetting codes are imbedded directly in the manuscript, you lose the flexibility of changing your mind about the typographic parameters without *detailed code changes*—as opposed to changing the extremely small number of entries in the typographic parameter table which you use when you use generic coding; and
- (2) If you're going to do something other than typesetting, for example having as a second output product form the provision of the content on a floppy disk, you definitely do not want, as Betty Davis mentioned this morning, to be stripping typesetters' tapes to get the codes out. That is an inversion that you would prefer not to do.

The point is (Figure 18) that these are data element or editorial element identifiers, and that's the significant thing. The next dimension is that they belong within an organized *editorial* structure of that particular document or that particular document title. The items standardized (Figure 19) are not the things you might expect, like the nuts and bolts and the codes, but, rather, the *methodology*. This is a content-driven situation, not a form-driven situation, and you absolutely must maintain the editorial discipline.

The standard that we will be leading into establishes (Figure 20) what you might call the syntactical legitimacy of the argument, and that, in essence, is a framework of allowability which, if you live by it, then you know you

can process this in machines large or small. You will need user code definitions, what you might call the side sheet or the head end of the tape, and you will need that inevitable preprocessing.

You need not be frightened by levels of implementation or levels of use if you're an author who works in a garret, because the Level 1 implementation can be done by anyone with just a microcomputer. If you go to the more powerful machines, you then can check this for validity against a given document type as to whether or not it has the proper structure and meets the requirements of syntax—you can check for it—all the things that are conditional upon maximizing the processing capability of large computer systems.

"Levels of use" (Figure 21) refers to the fact that you don't have to process everything in a given document by generic coding. You're going to hear some presentations about mathematical equations. The first-time user as an author can just make a space reservation, in essence a place-marker and the requisite area, for each of the figures to go in, for the equations to go in, for the illustrations to go in. Thus, generic coding can be used for a document that has drop-ins or for one that has plain text—with perhaps a half an hour of orientation. The suggestion behind the phrase "95/5" is that, in terms of tonnage of typesetting, 95% of it can be done reasonably simply; 5% takes more time. In the STM environment that is less true than it is in other environments, but unless you get the mixes of equations and mathematical symbols *in the running text* (as you will be exposed to later) this is still a truism.

What (Figure 22) is the distinction between machine readable, translatable, and transportable? Machine readable, you know. It's on some kind of magnetic medium. Translatable (Figure 23), you've heard about conversions and word processing. In essence, you go from Model A to Model X, and they're not the same. Whether they are both code machines, or one code and one space, they are not identical. You can't take the floppy from one and just walk over to the other as you really, truly should be able to do. If it were a transportable medium (Figure 24), you would be able to do that if the physical properties and the electrical properties were the same. As they exist, one word processor has the disk going one way and the other word processors have them going the other way. If you prepare your content with generic coding in it, you have to overcome the electrical translations between machines in any event; there is no escape from that.

But, *as an author, your basic goal is to preserve the integrity of your manuscript*. You have just content and generic coding. You can deal with the editorial content without any difficulty. Your *manuscript* is, indeed, transportable.

As to code standardization (Figure 25), we feel that you only need to know what the character coding is, what the control codes are, and what the generic codes are. It is not as important that they be one particular thing as long as you know what each is and you then can translate by table.

The symbols that are usually used for the delimiters (Figure 26) are the greater-than and less-than signs. They are used to identify the beginning and end of each uniquely different code you have for each different editorial element in the manuscript. If your machine does not have greater-

**AUTHOR-PREPARED
ELECTRONIC MANUSCRIPT**

Figure 1

WHY SHOULD

AN AUTHOR

A PUBLISHER

BE CONCERNED WITH

GENERIC CODING

TRANSPORTABLE FILES

Figure 3

AUTHORS'

MICROCOMPUTERS

AVAILABILITY

CONVENIENCE

EFFECTIVENESS

Figure 2

GENERIC CODING

MINIMIZES:

COST

VULNERABILITIES

AGGRAVATION

Figure 4

AUTHOR

PUBLISHER

TYPESETTER

1

4 2 1 3 5

Figure 5

WHAT IS

GENERIC CODING?

Figure 6

**GENERIC CODING
REQUIREMENTS
ARE
SYSTEM REQUIREMENTS**

Figure 7

**WHEN AN AUTHOR
CONCERNS HERSELF/HIMSELF
WITH OUTPUT FORM—E.G. TYPOGRAPHY,
SHE/HE BECOMES, FOR THE MOMENT,
A TYPOGRAPHIC DESIGNER**

Figure 9

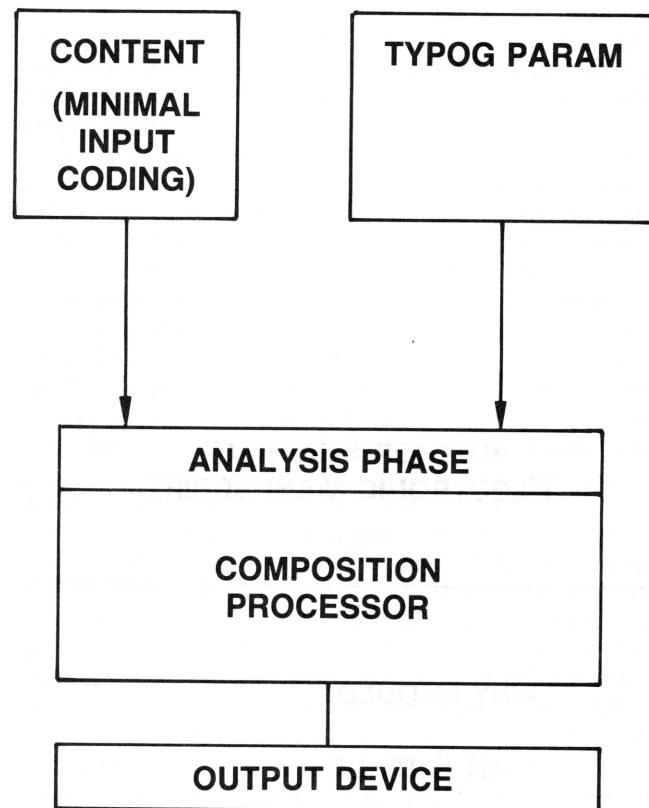


Figure 8

**SHOULD CODING FOR THE
OUTPUT PRODUCT FORM
BE MIXED WITH CODING FOR
INPUT CONTENT & CODING
NOT AT THE STAGE OF
CREATIVE MANUSCRIPT**

Figure 10

COEXISTENT
STRUCTURES
WYSIWYG
WYCIWYG

Figure 11

WYSIWYG

WHAT

YOU

SEE

IS

WHAT

YOU

GET

Figure 12

WYCIWYG
WHAT
YOU
CODE
IS
WHAT
YOU
GET

Figure 13

Emphasis on Secondary Education

The group structured its approach by distinguishing between probable and possible developments in the next ten years. What will the average secondary school *probably* look like a decade hence? This estimate was arrived at mainly by extrapolation of trends now visible in such respects as numbers of students and teachers, birth rates, years of formal schooling, and growth of both gross national product and educational budgets. The estimate excludes the possibilities of a world as it might be if a third world war intervened in the next decade or of a nation as it might be if our goals changed so radically as, for example, to shift the totality of the defense budget to educational purposes.

What could the average secondary school *possibly* look like a decade hence? In contrast to an assessment of probability, an estimate of possibility inevitably involves a measure of speculation and normative judgment. It must take account, moreover, both of tech-

Figure 14

Control and Manipulation of Behavior

Techniques for the alteration and control of human behavior, character, and personality, which have long been favorite devices for such science-fiction anti-utopias as *Brave New World* and *1984*, are today subjects of serious and respectable scientific research. Over the last decade, in particular, the technology of behavioral control—by physical, chemical, and psychological means—has made impressive advances, although it is the consensus of knowledgeable experts that the time is still distant when we will have the capability to apply such techniques on a large scale.

The social implications of the technical possibility of controlling human behavior might be suggested by such hypothetical cases as the following: If, as José Delgado of Yale is showing, an electrode introduced into the brain can be used to alter the aggressive behavior of an individual by remote control, or if psychochemical therapy can alter the behavior of an alcoholic or a narcotics addict, should not individuals and society be able to take advantage of such new possibilities to ameliorate personal misfortune or aid in the reduction of crime? It is not clear that the introduction of therapeutic procedures to enable, or even to force, actual and potential criminals to behave in a socially acceptable manner is necessarily a less desirable alternative than to continue to expand our police forces and populate our jails. Nor is it clear *a priori* that deliberate control of deviant behavior is worse than the personal and family tragedies that it often leads to.

Figure 15

THE ART OF CROSS-EXAMINATION

Fortunately, the plaintiff was not well informed on the law. His attention apparently had not been called to the decisions in *Morse v. Douglas*, 112 App. Div. 798 and *Gainsburg v. Bachrack*, 241 App. Div. 28, aff'd 266 N. Y. 468. So, on cross-examination, the plaintiff testified that he was told by the defendant that this was a new corporation that was about to be organized; that the defendant said that he would be an important member of the executive force of the

Figure 16

EDITORIAL CODES
TYPESETTING CODES
IS TYPESETTING THE
ONLY OUTPUT FORM?

Figure 17

GENERIC CODES
ARE
DATA ELEMENT IDENTIFIERS
WITHIN THE
EDITORIAL HIERARCHY

Figure 18

WHAT IS STANDARDIZED?

THE METHODOLOGY

CONTENT—DRIVEN

***NOT* FORM—DRIVEN**

THE EDITORIAL DISCIPLINE

Figure 19

SYNTACTICAL LEGITIMACY

USER CODE DEFINITIONS

PRE-PROCESSOR

Figure 20

LEVELS OF IMPLEMENTATION

LEVELS OF USE

Figure 21

WHAT IS A:

MACHINE-READABLE

TRANSLATABLE

TRANSPORTABLE

FILE?

Figure 22

TRANSLATABLE

**WORD
PROC'G
SOURCE**

CONVERSION

**WORD
PROC'G
DESTINATION**

NOT IDENTICAL

Figure 23

TRANSPORTABLE

**CONTENT
&
GENERIC
CODING**

**CONTENT
&
GENERIC
CODING**

IDENTICAL

Figure 24

CODE STANDARDIZATION

CHARACTER	CODES
CONTROL	CODES
GENERIC	CODES

Figure 25

WHAT IS A GENERIC CODE

- (GC) This is a representative generic code.
- (C1) Chapter Twenty-Five
- (H1) First-level Head
- \$P\$ Paragraph
- (IS) Index Word or Phrase Starting Here
- (EE) Emphasis Ends Here

Figure 26

SHOULD GENERIC CODES BE

FIXED

VARIABLE

STANDARDIZED

Figure 27

**HOW SHOULD
CODE DEFINITIONS
BE ORIENTED?**

TO THE USER

**TO THE DOCUMENT
KIND**

Figure 28

THERE IS

NO ESCAPE

FROM THE

INEVITABLE

PRE-PROCESSOR

Figure 29

WHAT IS AN

IMPLEMENTATION

EXAMPLE

Figure 30

SYSTEM

BROAD CONCEPT

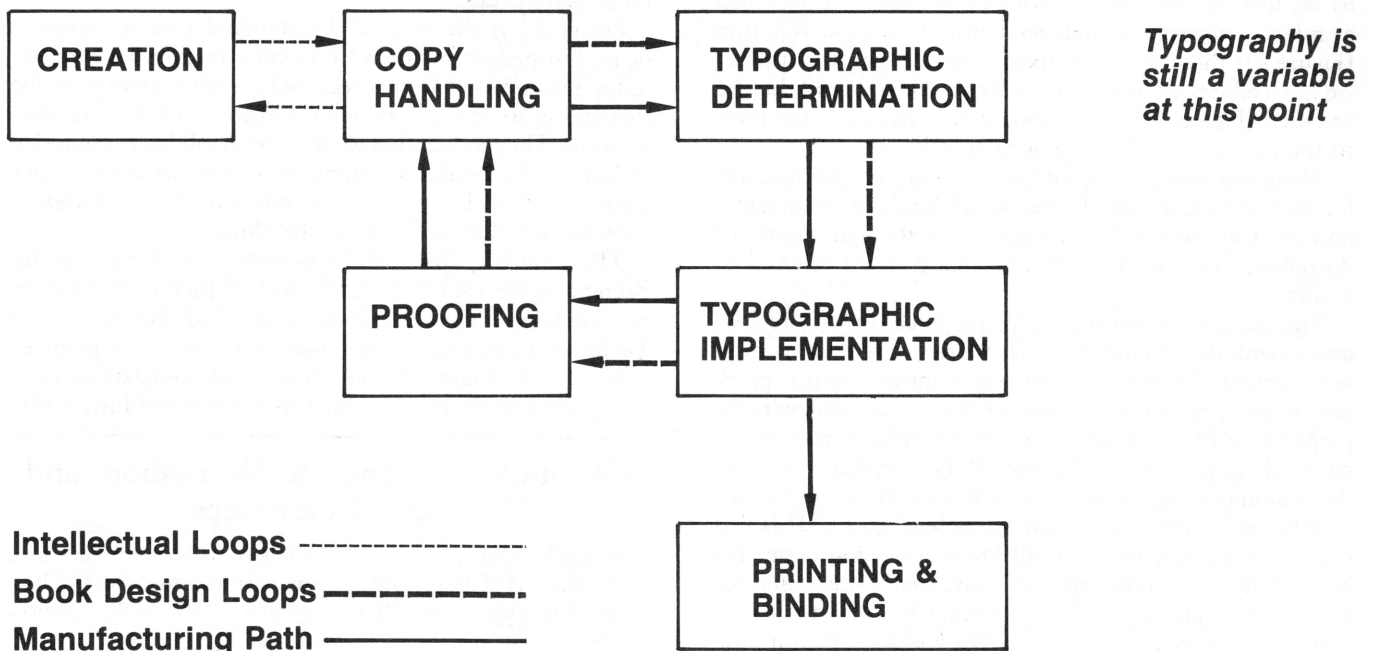


Figure 31

X3J6 STANDARD

SUBMISSION TO X3	JUNE 83
INTERNATIONAL REVIEW & COMMENT	JUNE 83
ISO TC97 SC5	SEPT 83
FINAL ACCEPTANCE	JULY 84

Information Processing Systems—
 Programming Languages—
 Text Interchange and Processing—
 Part Six: Document Markup Metalanguage

Figure 32

than or less-than signs, you can use a dollar sign or some other sign that is very low frequency in your manuscript. And, if you have to use that symbol, say, the dollar sign is somewhere in the manuscript, you can use the technique of the double strike to make it real, the single strike to make it a delimiter. We feel that *delimiters should be specifiable by the user*, in conjunction with his publisher, if he's had a project conference, or independently on his own. Whether (Figure 27) these should be fixed, variable, or standardized, we have addressed immediately above. How should they be oriented (Figure 28)? They should be oriented to the user, to the operator, and to the document kind.

There are manuscripts for books. There are manuscripts for technical manuals. There are all kinds of documents, and each industry will ultimately have its own registry of document kinds to allow the checking that I referred to earlier.

This leads into that inevitable preprocessor (Figure 29), and a reminder (Figure 30), that we're still separating form and content. In fact, the whole bottom line of this proposition and practice is that if you have independence in the preparation of the manuscript, as we believe you should have, then you can go to any of the suppliers, and the dimension of *service* is going to be what's important to you.

What is an implementation example (Figure 31)? Is this real? Is it being done? We all know it's being done. It's being done "each unto his own", and that's one reason for having the standard so that you don't get into too many variants, and, more particularly, that you don't get into the variant that ignores the editorial basis. There is a *typesetting* generic code conversation and use going on. At the same time there is an *editorial* generic code practice going on, and it's important when you talk to a typesetter to find out exactly which process he uses. This hospital supply catalogue example (Slide) indicates that it doesn't have to be running text. It also symbolizes the fact that you can use keyboard shorthand going in and that you should not confuse keyboard shorthand with the coding that you will use in a transportable file. The transportable file wants to be capable of being read in standard form by the person to whom you send it, whether it be a publisher or a typesetter, and this simply symbolizes that material that is not running text is just as easily taken care of.

There is also the business of the typesetters who will tell you that their codes are proprietary codes and they can't provide them to you. At our Graphic Communications Association meeting here last week, we had a paper pre-

sented where the man who was locked in the ice by proprietary codes took the bull by the horns, paid the price to have the magnetic tape decoded, put back into a GenCode[™] form, then put it out for bid. He got a sufficient reduction in the prices for his typesetting that it more than paid for the analysis work that he had to have done on the so-called proprietary codes.

Figure 32 is the title of the standard that is currently being finished off by the X3J6 Committee. Two weeks ago today (May 6, 1983) this received an affirmative vote for forwarding to the International Organisation for Standardisation. The product that we have been talking about today is Part 6. This establishes standard requirements for a Document Markup Language often referred to as a Standard Generalized Markup Language (SGML).

The schedule shows that this will go to ANSI, as the Secretariat for ISO TC97/SC5, in mid-June and will then be circulated for international review and comment. ISO Technical Committee 97 will meet in Ottawa in September to give this first consideration, and a projected date for final acceptance might be something in the order of July, 1984.

Chronological Speech, Workshop, and Paper References

Tunncliffe, William W. "Markup Language Concept." Paper read at the meeting of the Graphic Communications Association Character Generation Committee, September 1967, Hull, Quebec, Canada.

"Markup Language." Workshops sponsored by the Graphic Communications Association Character Generation Committee, July, August, and September 1968, Bethesda, Maryland.

Tunncliffe, William W. "Markup Language Concept." Paper read at the Annual Fall Conference of the Graphic Communications Association, November 1968, Las Vegas, Nevada.

Tunncliffe, William W. "Markup Language Concept." Paper read at the Annual Fall Conference of the Graphic Communications Association, November 1969, New Orleans, Louisiana.

Tunncliffe, William W. 1971. "System X—A User Need." In *Proceedings of the ASIS Workshop on Computer Composition* (Workshop held December 8-9, 1970 in Washington, D.C.), ed. Robert M. Landau, pp. 16-65. Washington, D.C.: American Society for Information Science.

Tunncliffe, William W. "Generic Coding Implementation Examples: Scott Stamp Catalogue and Harleco Hospital Supply Catalogue." Paper read at Annual Spring Conference of the Graphic Communications Association, May 1980, Atlanta, Georgia.